

comprising the steps of:

determining the set of files to be stored locally on the computer;  
identifying a type for each file in the set of files;  
storing the set of files locally on the computer;  
for each file, associating the type with a handler routine;  
sending each file to the associated handler routine to identify application  
functionality needed to run each file; and  
installing the identified application functionality locally on the computer.

5. (Original) The method of Claim 4, wherein the step of determining the set of files to be stored locally on the computer comprises receiving user input, wherein the user input corresponds to a plurality of files that are to be stored locally on the computer.

6. (Original) The method of Claim 4, wherein the step of determining the set of files to be stored locally on the computer comprises the steps of:  
searching a plurality of files in a plurality of storage locations on the computer;  
determining whether each file found in the plurality of storage locations is to be stored locally on the computer; and  
if so, then adding the file to the set of files.

7. (Canceled)

8. (Original) The method recited in Claim 4, wherein the handler routine comprises instructions for scanning the associated file and determining the application functionality that is needed to execute the associated file.

9. (Original) The method recited in Claim 8 wherein application functionality comprises programs, features and components.

10. (Amended) A method for identifying a set of application functionality to be stored

on a computer connected to a network, comprising the steps of:

causing a document identification engine (DIE) to create a list of a plurality of files stored locally on the computer;

sending the list of files from the DIE to a document mapping engine (DME);

causing the DME to identify a proper handler routine for each file in the list of files;

sending each file from the DME to the proper handler routine;

causing the handler routine to identify the application functionality needed to execute each file when the computer is disconnected from the network;

sending a list of needed application functionality of the handler routine to the DME;

sending a list of needed application functionality from the DME to a migration engine (ME);

causing the ME to determine the current status of the needed application functionality; and

if the status of the needed application functionality indicates that the needed application functionality is not installed locally on the computer, then causing the ME to install the needed application functionality to the computer.

11. (Original) A computer-readable medium comprising computer-readable instructions, which when executed, performs the steps of Claim 10.

12. (Original) The method of claim 6 wherein the step of determining whether each file found in the plurality of storage locations is to be stored locally is based on a set of rules.

13. (Original) The method of claim 6 wherein the step of determining whether each file found in the plurality of storage locations is to be stored locally is based on a user's usage patterns.

14. (Original) The method of claim 4 wherein the step of identifying application

functionality<sup>1</sup> needed to run each file comprises determining whether each file needs multiple application functionality.

15. (Original) The method of claim 14 wherein the step of determining whether each file needs multiple application functionality comprises mapping application functionality to a file embedded in a file in the set of files.

16. (Original) The method of claim 15 wherein the embedded file is an Object Linking and Embedding (OLE) object.

17. (Original) The method of claim 15 wherein the embedded file is a hyperlink.

18. (Amended) The method of claim 10 further comprising the steps of:  
causing the handler routine to notify the DME of an embedded file; and  
in response to receiving the notification of the embedded file, causing the DME to transmit the embedded file to another handler routine associated with the embedded file.

19. (Amended) The method of claim 10 further comprising the steps of:  
sorting the application functionality according to a frequency of occurrence.

20. (Amended) The method of claim 19 wherein the step of sorting the application functionality comprises steps of:  
causing the handler routine to return importance rankings associated with the application functionality.

21. (New) A method of executing application functionality on a computer disconnected from a network comprising:  
while the computer is connected to the network, identifying one or more files to be used when the computer is disconnected from the network;  
determining whether the one or more files are associated with the application

functionality, wherein the application functionality is located on the network;

upon identifying one or more files associated with the application functionality,  
transferring the application functionality from the network to the computer;

disconnecting the computer from the network; and

executing the application functionality in combination with at least one of the files  
associated with the application functionality.

22. (New) A method as defined in claim 21 wherein the one or more files are on the  
computer prior to the identifying act.

23. (New) A method as defined in claim 21 wherein the one or more files are on the  
network prior to the identifying act, the method further comprising:

prior to disconnecting the computer from the network, transferring the one or  
more files from the network to the computer.

24. (New) A method as defined in claim 21 wherein the act of determining whether  
the one or more files are associated with the application functionality comprises analyzing the  
one or more file to determine application functionality

25. (New) A method as defined in claim 24 further comprising:

receiving a disconnect signal; and

in response to the disconnect signal, automatically identifying the one or more  
files to be analyzed.

26. (New) A method as defined in claim 25 wherein the disconnect signal comprises a  
shut-down signal.

27. (New) A method as defined in claim 25 wherein the disconnect signal comprises a  
undock signal.